Available online at www.sciencedirect.com
SCIENCE DIRECT
DISCRETE APPLIED MATHEMATICS
Discrete Applied Mathematics 143 (2004) 182–193
www.elsevier.com/locate/dam

Multi-terminal multipath flows: synthesis[☆]

R. Chandrasekaran[a], K.P.K. Nair[b], Y.P. Aneja[c,*], S.N. Kabadi[b]

[a]Department of Computer Science, University of Texas at Dallas, Richardson, TX, USA
[b]Faculty of Administration, University of New Brunswick, Fredericton, NB, Canada E3B 5A3
[c]Faculty of Business Administration, University of Windsor, Windsor, Ont., Canada N9B 3P4

Received 20 September 2002; received in revised form 12 January 2004; accepted 26 February 2004

Abstract

Given an undirected network $G \equiv [N,E]$, a source–sink pair of nodes (s,t) in N, a non-negative number $u_{i,j}$ representing the capacity of edge (i,j) for each $(i,j) \in E$, and a positive integer q, an "elementary q-path flow" from s to t is defined as a flow of q units from s to t, with one unit of flow along each path in a set of q edge-disjoint s–t paths. A q-path flow from s to t is a non-negative linear combination of elementary q-path flows from s to t. In this paper we provide a strongly polynomial combinatorial algorithm for designing an undirected network with minimum total edge capacity which is capable of meeting, non-simultaneously, a given set of symmetric q-path flow requirements between all pairs of nodes. This extends the previous work on network synthesis.

Keywords: Network synthesis; Multipath flows; q-capacity of a cut; Strongly polynomial algorithm

1. Introduction

Network flow analysis and synthesis [3] are of considerable interest in communication. For improving reliability of communication flow between a source–sink pair, Kishimoto and Takeuchi [14] introduced the maximum q-path flow problem. Related results appear in [16,15]. In [13], Kishimoto presented a max-flow min-cut theorem and a strongly polynomial algorithm for finding a maximum q-path flow for both directed and undirected networks. Aggarwal and Orlin [1] gave additional results and an application.

In this paper we consider synthesis of an undirected network, with minimum sum of edge-capacities, for meeting, non-simultaneously, specified q-path flow requirements between every pair of nodes. After establishing certain properties of interest, we present a strongly polynomial algorithm for the above synthesis problem. Next section describes the q-path flow problem and the associated max-flow min-cut theorem. The synthesis problem, the algorithm, and illustrative examples are given in Section 3. Finally, Section 4 provides proofs that the algorithm yields an optimal solution to the synthesis problem.

2. Multipath flow problem

Given an undirected network $G \equiv [N,E]$, a source–sink pair (s,t) of nodes in N, a non-negative number $u_{i,j}$ representing the capacity of edge (i,j) for each $(i,j) \in E$, and a positive integer q, we define an "elementary q-path flow" from s

[☆] This work was supported by research grants from the Natural Sciences and Engineering Research council of Canada.
* Corresponding author.
E-mail address: aneja@uwindsor.ca (Y.P. Aneja).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2004.02.014

to t to be a flow of q units, with one unit of flow along each path in a set of q edge-disjoint paths from s to t. In an elementary q-path flow, at least one unit of flow is guaranteed if no more than $(q-1)$ edges fail. A q-path flow from s to t is a non-negative linear combination of elementary q-path flows from s to t, adhering to edge-capacities. Alternately, a q-path flow can be defined over node-disjoint paths. However, we shall confine to edge-disjoint paths. In a q-path flow of value F/q the value of the associated total flow from s to t is F. The objective is to obtain a maximum q-path flow, i.e. a q-path flow with maximum flow value. Kishimoto [13] provides a strongly polynomial algorithm for finding such a flow.

We now define q-capacity of a cut $(S,\bar{S})$ for any non-empty proper subset S of N. (This definition is implicit in [13].) Let $[e_1,e_2,\dots,e_m]$ be an ordering of the edges of the cut so that the capacities are non-increasing. Let $\alpha_k=\sum_{j=k}^{m}u_{e_j}$; $k=1,2,\dots,q$. Then the q-capacity of $(S,\bar{S})$ is given by the relation $\beta_q(S,\bar{S})=\min_{1\leqslant j\leqslant q}[\alpha_j/(q-j+1)]$. (If $m<q$, the q-capacity of the cut is zero.) For example,

$$\beta_2(S,\bar{S})=\min\left[\frac{\sum_{j=1}^{m}u_{e_j}}{2},\sum_{j=2}^{m}u_{e_j}\right]$$

and

$$\beta_3(S,\bar{S})=\min\left[\frac{\sum_{j=1}^{m}u_{e_j}}{3},\frac{\sum_{j=2}^{m}u_{e_j}}{2},\sum_{j=3}^{m}u_{e_j}\right].$$

Theorem 1 (Kishimoto [13]). *Let F/q be the value of any q-path flow from s to t and let $(S,\bar{S})$ be any cut separating s and t. Then $F/q\leqslant\beta_q(S,\bar{S})$. Moreover,*

$$\frac{F^*}{q}=\min_{\substack{S:\\ s\in S\\ t\in\bar{S}}}[\beta_q(S,\bar{S})],$$

where F^*/q is the maximum value of q-path flow.

3. Multi-terminal multipath flow network synthesis

We are given two positive integers q, n and non-negative numbers $r_{i,j}$; $1\leqslant i,j(\neq i)\leqslant n$ (satisfying the relation $r_{i,j}=r_{j,i}$), where $r_{i,j}$ represents the required value of q-path flow from node i to node j in an undirected network on the node set $N=\{1,2,\dots,n\}$. We want to determine the values of capacities $\{u_e\colon e\in E(K_n)\}$ on the edges of the complete undirected graph K_n, so that each of these requirements can be realized (one at a time) and the sum of all edge capacities is minimum.

This is a generalization of similar problem for regular maximum flows [3] (the case where $q=1$) considered by Mayeda [17], and Gomory and Hu [6]. Synthesis problem to design a network with minimum number of edges has been considered by Talluri [19]. The problem of synthesizing a network with minimum weighted sum of edge capacities, though the problem has polynomial time complexity, solution method involves solution of a large scale linear program [7] and only for a special case, in which the underlying network is restricted to be a cycle, a strongly polynomial combinatorial algorithms has been reported [9]. The integer version of synthesizing a network with minimum sum of edge capacities has been considered in [2,4,8,10,18]. For hop constrained flows, a strongly polynomial algorithm for synthesis problem to minimize the unweighted sum of edge capacities for meeting stated flow requirements is given by Gibbens and Kelley [5]. For integer version of synthesis problem for 3-hop flows—a strongly polynomial algorithm when the requirements are large is given in [12].

Let $\pi_i:=\max_{j\neq i}[r_{i,j}]$. It is easy to see that $r_{i,j}\leqslant\min[\pi_i,\pi_j]$ for all i and j. Our network will allow possibly larger values of $\min[\pi_i,\pi_j]$ and still have the least total capacity similar to the Gomory–Hu solution for the case $q=1$. The problem can be formulated as

$$\begin{aligned}&\min\quad\left[\sum_{e\in E(K_n)}u_e\right]\\ &\text{s.t.}\quad \beta_q(S,\bar{S})\geqslant\max_{\substack{i\in S\\ j\in\bar{S}}}[r_{i,j}]\quad\forall S\subset N,\quad u_e\geqslant 0\quad\forall e\in E(K_n).\end{aligned}\tag{LP1}$$

Problem (LP1) is equivalent to the following linear program:

$$\begin{aligned}
&\min \quad \left[\sum_{e\in E(K_n)} u_e\right]\\
&\text{s.t.} \quad \frac{1}{q-k+1}\sum_{j=k}^{m} u_{e_j} \geqslant \max_{\substack{i\in S\\ j\in \bar S}}[r_{i,j}] \quad \forall S\subset N;\ 1\leqslant k\leqslant q \text{ and all orderings } [e_1,e_2,\ldots,e_m]\\
&\qquad\qquad \text{of the edges of the cut } (S,\bar S)\ u_e\geqslant 0 \quad \forall e\in E(K_n).
\end{aligned} \tag{LP2}$$

We will produce a set of lower bounds for the optimal objective function value in linear program (LP2) as follows. By considering only some of the constraints in (LP2) corresponding to sets of the form S with $|S|=1$, we will define the following linear program (LP3). We will then obtain bounds for the optimal objective function value in (LP3).

Let the elements of N be ordered so that $\pi_1=\pi_2\geqslant\pi_3\geqslant\cdots\geqslant\pi_n$. For any $i\in N$ and $k\leqslant i$, let

$$E_{i,k}=\{(i,j)\colon\ j\geqslant k, j\neq i\},$$

$$E_k=\bigcup_{i=k}^{n}E_{i,k}=\{e\in E(K_n)\colon\ e=(i,j);\ i,j\geqslant k, i\neq j\}.$$

The linear program (LP3) is as follows:

$$\begin{aligned}
&\min \quad \left[\sum_{e\in E(K_n)} u_e\right]\\
&\text{s.t.} \quad \frac{1}{q-k+1}\sum_{e\in E_{i,k}} u_e\geqslant \pi_i \quad \forall i\in N;\ 1\leqslant k\leqslant \min[i,q], \quad u_e\geqslant 0 \quad \forall e\in E(K_n).
\end{aligned} \tag{LP3}$$

We shall refer to the constraint of (LP3) corresponding to given values of i and k as the (i,k)th constraint. This eases our burden in referring to these many times.

3.1. Lower bounds

Our strategy is to identify a set of lower bounds on the optimal objective function value of the linear program (LP3) and develop an algorithm that produces a feasible solution to the synthesis problem that achieves one of these lower bounds. This will imply that the solution produced by the algorithm is optimal for the synthesis problem.

Lemma 2. *Each of the following is a valid lower bound for the optimal objective function value of the linear program* (LP3):

$$LB_p^{(q)}(\pi)=\sum_{j=1}^{q-p+1}(q-j+1)\pi_j+\frac{(p-1)}{2}\xi_{q-p+2}$$

for $p=1,2,\ldots,q+1$, *where* $\xi_j=\sum_{i=j}^{n}\pi_i$.

Proof. For any $p=1,2,\ldots,q+1$, we prove the validity of $LB_p^{(q)}(\pi)$ as follows:

(a) Add constraints in the set $\{(i,q-p+2)\colon\ q-p+2\leqslant i\leqslant n\}$ and multiply the resultant inequality by $(p-1)/2$ to obtain

$$\sum_{e\in E_{q-p+2}}u_e\geqslant\frac{p-1}{2}\xi_{q-p+2}.$$

(b) For each $j=1,2,\ldots,q-p+1$, multiply the constraint (j,j) by $(q-j+1)$ and add them all to the above inequality to get

$$\sum_{e\in E(K_n)}u_e\geqslant\sum_{j=1}^{q-p+1}(q-j+1)\pi_j+\frac{p-1}{2}\xi_{q-p+2}=LB_p^{(q)}(\pi).$$

For example, for $p = q+1$, add the set of constraints in the group $(i,1)$ over all $i \in N$, and multiply the resultant inequality by $q/2$ to obtain

$$\sum_{e \in E(K_n)} u_e \geqslant \frac{q}{2} \sum_{i=1}^{n} \pi_i = \frac{q}{2}\, \xi_1 = LB^{(q)}_{q+1}(\pi).$$

For $p = q$:

(a) Add constraints of the form $\{(i,2)\colon 2 \leqslant i \leqslant n\}$, and multiply the resultant inequality by $(q-1)/2$ to get

$$\sum_{e \in E_2} u_e \geqslant \frac{q-1}{2}\, \xi_2.$$

(b) Multiply constraint $(1,1)$ by q and add to the above to get

$$\sum_{e \in E(K_n)} u_e \geqslant q\pi_1 + \frac{q-1}{2}\, \xi_2 = LB^{(q)}_{q}(\pi)$$

This completes the proof. □

Let

$$\Delta^{(q)}_p(\pi) = \begin{cases} LB^{(q)}_p(\pi) - LB^{(q)}_{p+1}(\pi), & 1 \leqslant p \leqslant q, \\ \dfrac{p\pi_{q-p+1} - \xi_{q-p+2}}{2}, & 1 \leqslant p \leqslant q. \end{cases}$$

Lemma 3. $\Delta^{(q)}_p(\pi) \geqslant 0 \Rightarrow \Delta^{(q)}_{p+1}(\pi) \geqslant 0,\ 1 \leqslant p \leqslant q-1.$

Proof.

$$\Delta^{(q)}_{p+1}(\pi) - \Delta^{(q)}_p(\pi) = \frac{p+1}{2}\,(\pi_{q-p} - \pi_{q-p+1}) \geqslant 0, \qquad 1 \leqslant p \leqslant q-1.$$

This completes the proof. □

Let

$$p^* = \min[\{q+1\} \cup \{p\colon \Delta^{(q)}_p(\pi) > 0\}].$$

Then, Lemma 3 implies that $LB^{(q)}_{p^*}(\pi) = \max_p LB^{(q)}_p(\pi)$. This identifies the strongest of the bounds for a particular instance of the problem. Moreover, we get the following corollary from Lemma 3:

Corollary 1. $\pi_{q-p} = \pi_{q-p+1} \Rightarrow p^* \neq p+1$. *In particular, since* $\pi_1 = \pi_2$, $p^* \neq q$, *and hence,* $LB^{(q)}_q(\pi)$ *can be ignored.*

Proof. $(\pi_{q-p} = \pi_{q-p+1}) \Rightarrow (\Delta^{(q)}_{p+1}(\pi) - \Delta^{(q)}_p(\pi) = 0) \Rightarrow LB^{(q)}_{p+1}(\pi) = (LB^{(q)}_p(\pi) + LB^{(q)}_{p+2}(\pi))/2$. Hence $LB^{(q)}_{p+1}(\pi)$ is no bigger than either $LB^{(q)}_p(\pi)$ or $LB^{(q)}_{p+2}(\pi)$. □

3.2. Algorithm

Input: Nonnegative numbers $r_{i,j}$; $1 \leqslant i, j(\neq i) \leqslant n$ (satisfying the relation $r_{i,j} = r_{j,i}$), and a positive integer q, $2 \leqslant q < n$.
Output: Capacities $\{u_e\colon e \in E(K_n)\}$ of edges of the complete undirected graph K_n.
Feasibility: A set of values for $\{u_e\colon e \in E(K_n)\}$ is feasible if the q-path flow between nodes i and j is no less than $r_{i,j}$ for all i, j pairs (taken one pair of nodes at a time).
Optimality: A feasible solution is optimal if among all feasible solutions its value for $\sum_{e \in E(K_n)} u_e$ is minimum.

Let $\pi_i = \max_{j \neq i}[r_{i,j}]$. Then, after renumbering nodes (if needed), we have $\pi_1 = \pi_2 \geqslant \pi_3 \geqslant \cdots \geqslant \pi_n$. We will assume, from now on, that nodes are so numbered.
Initially $k = 0$; $\pi^{(0)}_i = \pi_i$ for all i.
In the kth iteration, let $\pi^{(k)} = [\pi^{(k)}_1, \pi^{(k)}_2, \ldots, \pi^{(k)}_{n_k}, 0, 0, \ldots, 0]$ where $\pi^{(k)}_1 = \pi^{(k)}_2 \geqslant \cdots \geqslant \pi^{(k)}_{n_k} > 0$.

There are two cases to consider:

Case 1: $n_k \geqslant q+1$: Let m_k and ℓ_k be, respectively, the smallest and largest indices such that $\pi^{(k)}_{m_k+1} = \pi^{(k)}_q = \pi^{(k)}_{\ell_k}$. If $\ell_k = n_k$, then set $\ell_k := n_k - 1$. Let $\Theta^{(k)} = \min\{\pi^{(k)}_{n_k}, \Theta^{(k)}_1, \Theta^{(k)}_2\}$, where

$$\Theta^{(k)}_1 = \begin{cases} \dfrac{\ell_k - m_k}{q - m_k}\,(\pi^{(k)}_{\ell_k} - \pi^{(k)}_{\ell_k+1}) & \text{if } \ell_k < n_k - 1, \\ \infty & \text{otherwise,} \end{cases}$$

$$\Theta^{(k)}_2 = \begin{cases} \dfrac{\ell_k - m_k}{\ell_k - q}\,(\pi^{(k)}_{m_k} - \pi^{(k)}_{m_k+1}) & \text{if } m_k > 0;\ \ell_k > q, \\ \infty & \text{otherwise.} \end{cases}$$

We create complete graphs on each node-set of the form $\{1,2,\dots,m_k,\ i_1,i_2,\dots,i_{q-m_k},n_k\}$ where $m_k+1 \leqslant i_1 < i_2 < \cdots < i_{q-m_k} \leqslant \ell_k$, and we assign to each edge of each of these graphs a capacity of

$$\frac{\Theta^{(k)}}{\binom{\ell_k - m_k}{q-m_k}}.$$

We change the π vector as follows:

$$\pi^{(k+1)}_i = \begin{cases} \pi^{(k)}_i - \Theta^{(k)} & 1 \leqslant i \leqslant m_k \text{ and } i = n_k, \\ \pi^{(k)}_i - \dfrac{q - m_k}{\ell_k - m_k}\Theta^{(k)} & m_k + 1 \leqslant i \leqslant \ell_k, \\ \pi^{(k)}_i & \text{otherwise.} \end{cases}$$

We superpose these graphs to obtain a graph $G^{(k)} \equiv [N^{(k)}, E^{(k)}]$ on node set $N^{(k)} = \{1,2,\dots,\ell_k,n_k\}$. The capacities of the edges of $G^{(k)}$ can be directly obtained as follows:

$$u^{(k)}_{i,j} = \begin{cases} \Theta^{(k)} & \text{if } \{i,j\} \subset \{1,2,\dots,m_k,n_k\}, \\ \dfrac{q-m_k}{\ell_k - m_k}\,\Theta^{(k)} & \text{if } i \in \{1,2,\dots,m_k,n_k\} \text{ and } j \in \{m_k+1,\dots,\ell_k\}, \\ \dfrac{(q-m_k)(q-m_k-1)}{(\ell_k-m_k)(\ell_k-m_k-1)}\,\Theta^{(k)} & \text{if } m_k < (q-1) \text{ and } \{i,j\} \subset \{m_k+1,\dots,\ell_k\}. \end{cases}$$

Case 2: $n_k \leqslant q$: We create a graph $G^{(k)} \equiv [N^{(k)}, E^{(k)}]$ on the node set $N^{(k)} = \{1,2,\dots,q+1\}$ with edge set $E^{(k)} = \{(i,j)\colon 1 \leqslant i \leqslant n_k;\ 1 \leqslant j \leqslant q+1\}$ and assign to each edge $(i,j) \in E^{(k)}$ a capacity $u^{(k)}_{i,j}$ equal to $\max[\pi^{(k)}_i, \pi^{(k)}_j]$. We set $\pi^{(k+1)} = 0$ and stop.

Our final solution, $G^\star \equiv [N, E^\star]$, is obtained by superposing all the graphs $G^{(0)}, G^{(1)}, \dots, G^{(k^\star)}$, generated by the algorithm at various iterations. Thus, $E^\star$ is the union of the edge sets $E^{(0)}, E^{(1)}, \dots, E^{(k^\star)}$. The capacity, u_e of each edge $e \in E^\star$ is the sum of capacities assigned to this edge in $G^{(0)}, G^{(1)}, \dots, G^{(k^\star)}$. For every edge $e \in E(K_n) - E^\star$, $u_e = 0$.

Remark 1. The condition $[\pi^{(k)}_1 = \pi^{(k)}_2 \geqslant \pi^{(k)}_3 \geqslant \cdots \geqslant \pi^{(k)}_{n_k}]$ is always preserved; moreover, if $\pi^{(k)}_j = \pi^{(k)}_{j+1}$, then for any $t > k$, $\pi^{(t)}_j = \pi^{(t)}_{j+1}$ provided $\pi^{(t)}_{j+2} > 0$. Case 2 can occur only in the final iteration, $k^\star$, of the algorithm. If in the final iteration, we get Case 1, then we must have $[\pi^{(k^\star)}_1 = \pi^{(k^\star)}_2 = \cdots = \pi^{(k^\star)}_{n_{k^\star}}]$, and $n_{k^\star} = q+1$.

The algorithm description is now complete.

Example 1. $\pi = \pi^{(0)} = [20,20,7,5,4,2];\ q = 3$.

First step is of a Case 1 type; $n_0 = 6,\ m_0 = 2,\ \ell_0 = 3;\ \Theta^{(0)} = \pi^{(0)}_6 = 2$. Capacity increases are given by the uniform complete graph in Fig. 1:

Now $\pi^{(1)} = [18,18,5,5,4,0];\ n_1 = 5,\ m_1 = 2,\ \ell_1 = 4$, and we have $\Theta^{(1)} = \min[\pi^{(1)}_5, \frac{4-2}{4-3}(\pi^{(1)}_2 - \pi^{(1)}_3)] = \min[4,26] = 4$. Capacity increases at this step are given by the two graphs in Fig. 2:

$\pi^{(2)} = [14,14,3,3,0,0];\ n_2 = 4,\ m_2 = 2,\ \ell_2 = 4$. Again we have Case 1 instance with $\Theta^{(2)} = 3$, and capacity increase at this step is given by the Fig. 3 graph:

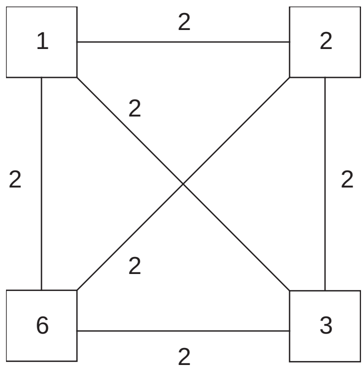

Fig. 1.

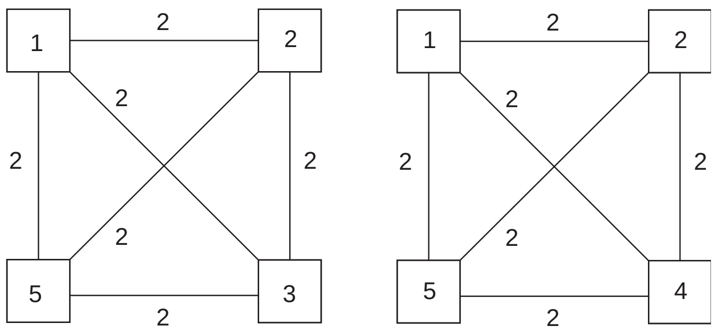

Fig. 2.

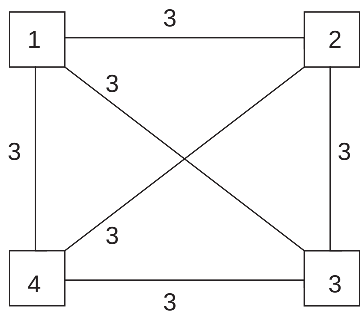

Fig. 3.

$\pi^{(3)} = [11, 11, 0, 0, 0, 0]$; $n_3 = 2$. Now Case 2 applies and capacity increases are given by the Fig. 4 graph:

$\pi^{(4)} = 0$. So, we stop.

The final network, $G^{\star}$, is obtained by superposing all these graphs. This achieves the lower bound $LB_2^{(3)}(\pi) = 109$. To show feasibility, we note that for the pair $(1, 2)$, we have 2 units of 3-path flow in Fig. 1, 2 units each in the two networks in Fig. 2, 3 units from Fig. 3, and 11 units from Fig. 4. These add up to the required 20 units of flow. Now for the pairs $(1, 3)$ or $(2, 3)$, we get 2 units from the first diagram, 2 units from the first diagram in the second set, and 3 units from the last diagram making up a total of 7. Similar arguments hold for pairs of the form $(1, i)$ or $(2, i)$; $i \geqslant 3$. For pairs like $(3, 4)$, we combine paths from 3 to 1 and those from 4 to 1.

Example 2. $\pi = \pi^{(0)} = [20, 20, 18, 16, 6, 4, 2]$; $q = 3$.

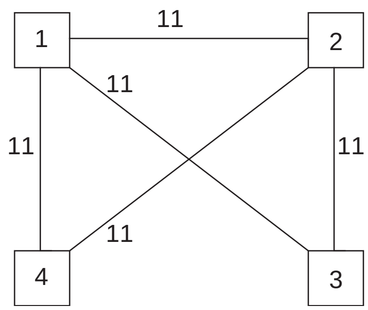

Fig. 4.

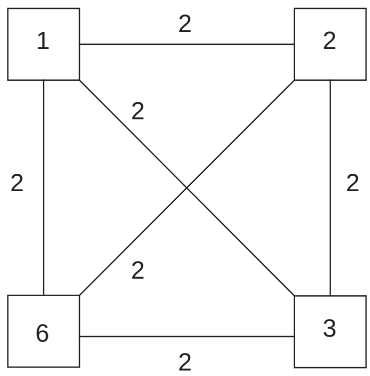

Fig. 5.

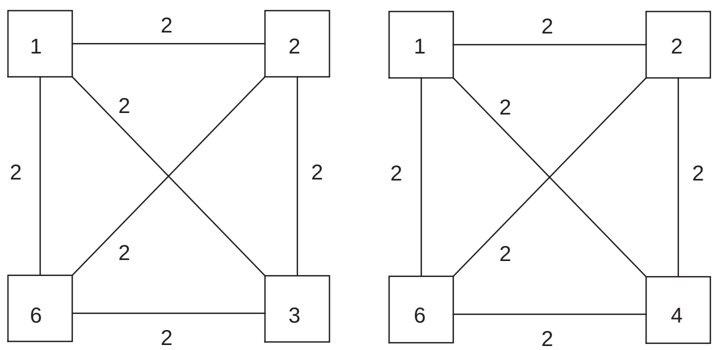

Fig. 6.

First step is of Case 1; $n_0 = 7$; $\Theta^{(0)} = \pi_7^{(0)} = 2$ (see Fig. 5):

$\pi^{(1)} = [18, 18, 16, 16, 6, 4, 0]$; $n_1 = 6$, $m_1 = 2$, $\ell_1 = 4$. Now we have a Case 1 instance and $\Theta^{(1)} = \min[\pi_6^{(1)}, \frac{4-2}{3-2}(\pi_4^{(1)} - \pi_5^{(1)}), \frac{4-2}{4-3}(\pi_2^{(1)} - \pi_3^{(1)})] = \min[4, 20, 4] = 4$. Capacity increases at this step are given by Fig. 6 graphs:

$\pi^{(2)} = [14, 14, 14, 14, 6, 0, 0]$; $n_2 = 5$; $m_2 = 0, \ell_2 = 4$. We have a Case 1 instance and capacity increases at this step are given by the four graphs in Fig. 7:

$\pi^{(3)} = [9.5, 9.5, 9.5, 9.5, 0, 0, 0]$; $n_3 = 4$. Again we have a Case 1 instance. Capacity increase at this step is given by the graph in Fig. 8.

Now $\pi^{(4)} = 0$ and we stop. We have the resulting network $G^{\star}$ by superposing all these graphs. This achieves the $LB_4^{(3)}(\pi) = \frac{q}{2}\,\xi_1 = 129$. Feasibility arguments are similar to those in the previous example.

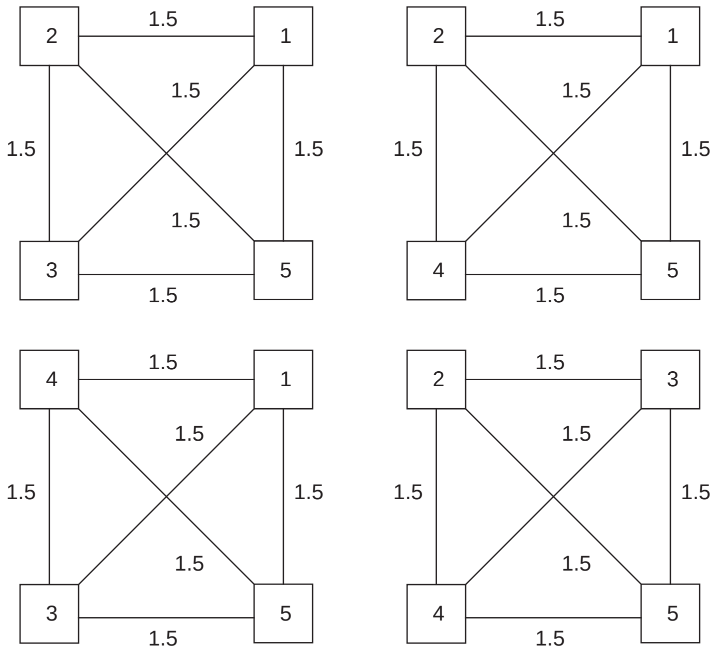

Fig. 7.

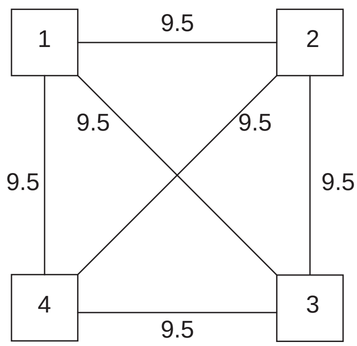

Fig. 8.

Example 3. $\pi = \pi^{(0)} = [20, 20, 10, 5, 2]$; $q = 3$. The three graphs, generated by the algorithm, in Fig. 9, superposed provide $G^{\star}$.

The value in this example is that of $LB_1^{(3)}(\pi) = 110$.

4. Validation of the algorithm

We shall now show that the algorithm terminates in strongly polynomial time with an optimal solution to the synthesis problem.

Lemma 4. *The algorithm terminates in* $\mathrm{O}(n^3)$ time.

Proof. Case 2 of the algorithm can occur only in the final iteration of the algorithm. In every iteration k of type Case 1, we have $m_{k+1} \leqslant m_k$, $\ell_{k+1} \geqslant \ell_k$, and $n_{k+1} \leqslant n_k$. In addition, at least one of the following holds: (a) $\pi_{n_k}^{(k+1)} = 0$; (b)

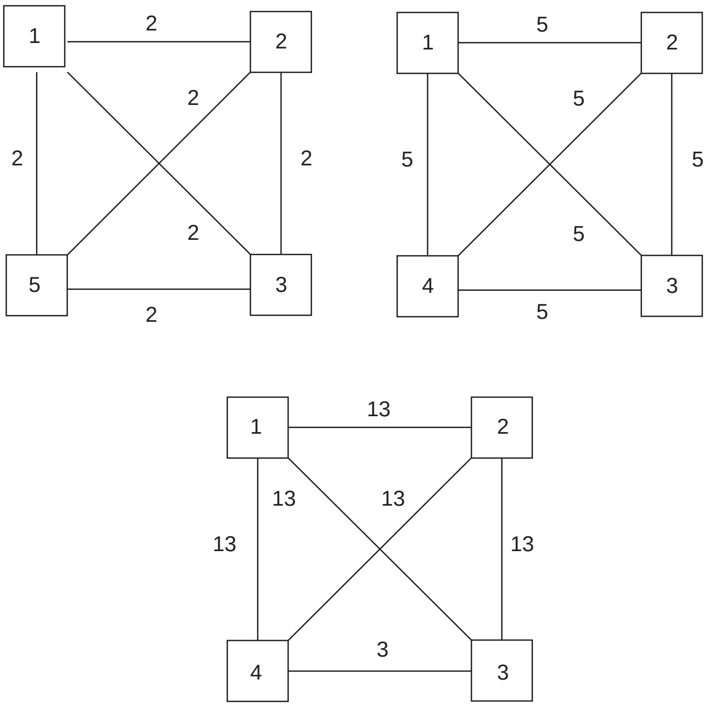

Fig. 9.

$\pi^{(k+1)}_{\ell_k} = \pi^{(k+1)}_{\ell_k+1}$ (this can occur only if $\ell_k < n_k - 1$); or (c) $\pi^{(k+1)}_{m_k} = \pi^{(k+1)}_{m_k+1}$ (this can occur only if $m_k > 0$). Hence, either we get $m_{k+1} < m_k$ or $\ell_{k+1} > \ell_k$ or $n_{k+1} < n_k$. Since $m_k < q\ \forall k$, it can decrease no more than q times. Since $q \leqslant \ell_k \leqslant n_k\ \forall k$, increase in value of ℓ_k and decrease in value of n_k can together take place no more than $n - q$ times. The total number of iterations of type Case 1 is thus at most n. In each iteration, k, the graph $G^{(k)}$ can be constructed in $O(n^2)$ time. The superposition of all the graphs $G^{(0)}, G^{(1)}, \ldots, G^{(k^\star)}$ will require $O(n^3)$ effort. This gives us the overall complexity of $O(n^3)$. □

Now, we shall prove that the network produced by the algorithm is feasible for the synthesis problem. For this, we show that for any $i, j \in N$, we can send $\min[\pi_i, \pi_j]$ units of q-path flow between nodes i and j in the network synthesized by the algorithm. We shall need the following two lemmas and corollary, the proofs of which are straightforward, and hence are omitted.

Lemma 5. *Consider any two undirected networks, G' and G'' on the same node set N. Let network $\bar{G}$ be obtained by superposing the two networks. Then for any cut $(S, \bar{S})$ on N, the sum of q-capacities of the cut in G' and G'' is no more than the q-capacity of the cut in $\bar{G}$.*

Lemma 6. *Let $v^\star_{i,j}, v^\star_{j,k}, v^\star_{i,k}$ represent maximum q-path flow values between pairs $(i,j), (j,k)$ and (i,k), respectively, in an arbitrary undirected network. Then, $v^\star_{i,k} \geqslant \min[v^\star_{i,j}, v^\star_{j,k}]$.*

Corollary 2. *Let G be an undirected network on n nodes and let $\pi = [\pi_1, \pi_2, \ldots, \pi_n]$ with $\pi_i \geqslant \pi_{i+1}$. If in G we can send π_i units of q-path flow between i and 1 for all i, then for any pair $\{i, j\}$ of nodes of G, we can send $\min[\pi_i, \pi_j]$ units of q-path flow between nodes i and j.*

Theorem 7. *The algorithm produces a feasible solution to the synthesis problem.*

Proof. Using Corollary 2, it suffices to prove that in the superposed graph, $G^\star$, output by the algorithm, we can send π_i units of q-path flow between i and 1 for all i. We shall prove the result by induction on the value of $k^\star$, the final iteration number of the algorithm.

If $k^\star = 0$, the algorithm requires only one iteration. In this case, the feasibility of the network produced can be easily seen. Suppose the result is true for all values of $k^\star < h$ for some $h > 0$. Let us now prove the result for $k^\star = h$.

Consider an instance of the problem for which $k^\star = h$. The algorithm generates vectors $\pi^{(0)}(=\pi), \pi^{(1)}, \pi^{(2)}, \ldots, \pi^{(h)}, \pi^{(h+1)}$ $(=0)$, and corresponding networks, $G^{(0)}, G^{(1)}, \ldots, G^{(h)}$. By induction argument, it follows that for any i, we can send $\pi_i^{(1)}$ units of q-path flow between i and 1 using the network obtained by superposition of the networks $G^{(1)}, \ldots, G^{(h)}$. Since, $h > 0$, the first iteration of the algorithm will be an instance of Case 1. In this iteration, we create complete graphs on each node set of the form $\{1, 2, \ldots, m_0, i_1, i_2, \ldots, i_{q-m_0}, n_0\}$ where $m_0 < i_1 < i_2 < \cdots < i_{q-m_0} \leqslant \ell_0$ and we assign to each edge of each of these graphs a capacity of

$$\frac{\Theta^{(0)}}{\binom{\ell_0 - m_0}{q - m_0}}$$

and this changes the $\pi^{(0)}$ vector as follows:

$$\pi_i^{(1)} = \begin{cases} \pi_i^{(0)} - \Theta^{(0)}, & 1 \leqslant i \leqslant m_0 \text{ and } i = n_0, \\ \pi_i^{(0)} - \dfrac{q - m_0}{\ell_0 - m_0}\, \Theta^{(0)}, & m_0 + 1 \leqslant i \leqslant \ell_0, \\ \pi_i^{(0)}, & \text{otherwise.} \end{cases}$$

Consider any $i \in \{2, 3, \ldots, n_0\}$.

If $i \in \{\ell_0 + 1, \ell_0 + 2, \ldots, n_0 - 1\}$ then $\pi_i^{(1)} = \pi_i^{(0)}$, and it follows by induction argument that we can send $\pi_i = \pi_i^{(1)}$ units of q-path flow between nodes 1 and i.

Suppose $i \in \{1, 2, \ldots, m_0\}$. Then in every one of the $\binom{\ell_0 - m_0}{q - m_0}$ graphs formed in the first iteration, we can send

$$\frac{\Theta^{(0)}}{\binom{\ell_0 - m_0}{q - m_0}}$$

units of q-path flow between nodes 1 and i. This gives a total q-path flow of value $\Theta^{(0)}$ between nodes 1 and i using only network $G^{(0)}$. The result now follows by induction argument and the fact that $\pi_i^{(1)} = \pi_i^{(0)} - \Theta^{(0)}$.

Suppose $i \in \{m_0 + 1, m_0 + 2, \ldots, \ell_0\}$. If $m_0 = q - 1$, then every cut in $G^{(0)}$ separating nodes 1 and i has at least q edges, each with capacity at least $\frac{(q-m_0)}{(\ell_0 - m_0)}\Theta^{(0)}$. If $m_0 < q - 1$, then every cut in $G^{(0)}$ separating nodes 1 and i has at least ℓ_0 edges, each with capacity at least $\frac{(q-m_0)(q-m_0-1)}{(\ell_0-m_0)(\ell_0-m_0-1)}\,\Theta^{(0)}$. Thus, in either case, the total q-capacity of every cut in $G^{(0)}$ separating nodes 1 and i is at least $\frac{(q-m_0)}{(\ell_0-m_0)}\Theta^{(0)} = \pi_i^{(0)} - \pi_i^{(1)}$. The result now follows by the max-flow-min-cut theorem (Theorem 1), and the induction argument.

So far, we have shown that we can send π_i units of q-path flow between nodes 1 and i for any $i < n_0$. Let us now consider the case $i = n_0$. It will be sufficient to show that every cut $(S, \bar{S})$ in $G^\star$, such that node 1 is in S and node n_0 is in $\bar{S}$, has a q-capacity of at least π_{n_0}. If $\bar{S}$ contains some node $j < n_0$, then the q-capacity of the cut is at least $\pi_j \geqslant \pi_{n_0}$, since the result has been shown to hold for every node $i < n_0$. So, let us consider the case, $\bar{S} = \{n_0\}$. The q-capacity of this cut in $G^{(0)}$ can be easily seen to be at least $\Theta^{(0)}$. The result now follows from the induction argument, Lemma 5, and the fact that $\pi_{n_0}^{(0)} - \pi_{n_0}^{(1)} = \Theta^{(0)}$. This proves the result. □

Theorem 8. *The network produced by the algorithm is optimal for the synthesis problem.*

Proof. We have already proved feasibility of the network in Theorem 7. We prove optimality by showing that the algorithm achieves one of the lower bounds developed in Section 3.

We compare, the values $\sum_i (\pi_i^{(k)} - \pi_i^{(k+1)})$ and $\sum_{e \in E^{(k)}} u_e^{(k)}$ in every iteration k of the algorithm.

If an iteration k is of type Case 1, then

$$\frac{\sum_{e \in E^{(k)}} u_e^{(k)}}{\sum_i (\pi_i^{(k)} - \pi_i^{(k+1)})} = \frac{q}{2}.$$

Hence, if Case 2 is not encountered, then $\sum_{e \in E(K_n)} u_e = \frac{q}{2}\,\xi_1$. This is equal to $LB_{q+1}^{(q)}(\pi)$.

Suppose Case 2 is encountered in the last iteration, k^*, and $\pi^{(k^*)} = [\pi_1^{(k^*)}, \pi_2^{(k^*)}, \ldots, \pi_{n_k}^{(k^*)}, 0, 0, \ldots, 0]$, where $\pi_{n_k}^{(k^*)} > 0$ and $n_{k^*} \leqslant q$. Then the $\pi^{(k^*-1)}$-vector has the form

$$[\pi_1^{(k^*-1)} = \pi_2^{(k^*-1)} > \pi_{m_{k^*-1}+1}^{(k^*-1)} = \cdots = \pi_q^{(k^*-1)} \geqslant \pi_{q+1}^{(k^*-1)}, 0, \ldots, 0].$$

If $\pi_q^{(k^*-1)} = \pi_{q+1}^{(k^*-1)}$, then $n_{k^*} = m_{k^*-1}$. If $\pi_q^{(k^*-1)} > \pi_{q+1}^{(k^*-1)}$, then $\ell_{k^*-1} = q = \ell_0$. This follows from noting that $m_0 \geqslant m_1 \geqslant \cdots \geqslant m_{k^*-1}$ and $q \leqslant \ell_0 \leqslant \ell_1 \leqslant \cdots \leqslant \ell_{k^*-1}$. Also note that at any iteration $t < k$, since Case 1 applies, $\pi_1^{(t)}, \pi_2^{(t)}, \ldots, \pi_{m_t}^{(t)}$ decrease by the same amount, whereas if $\ell_{k^*-1} = q$, then all $\pi_1^{(t)}, \pi_2^{(t)}, \ldots, \pi_q^{(t)}$ decrease by the same amount. Hence,

$$\pi_i^{(0)} - \pi_i^{(k^\star)} = \pi_j^{(0)} - \pi_j^{(k^\star)} \quad \forall 1 \leqslant i < j \leqslant n_{k^\star}.$$

Let

$$\pi_i^{(k^\star)} = \pi_i^{(0)} - \Theta \quad \forall 1 \leqslant i \leqslant n_{k^\star}.$$

In each of the iterations $k = 0, 1, \ldots, k^\star - 1$, each unit of reduction in $\pi_1^{(k)}$ is accompanied by $(q+1)$ units of reduction in $\sum_i \pi_i^{(k)}$, and $(q + 1 - n_{k^\star})$ units of reduction in $\sum_{i=n_{k^\star}+1}^{n} \pi_i^{(k)}$. Hence, the total increase in capacity in iterations $0, 1, \ldots, k^\star - 1$ equals $(q(q+1)/2)\Theta$. Also, since $\pi_{n_{k^\star}+1}^{(k^\star)} = 0$, it follows that

$$\Theta = \frac{\xi_{n_{k^\star}+1}}{q + 1 - n_{k^\star}}.$$

The increase in capacity in iteration $k^\star$ is given by

$$q\pi_1^{(k^\star)} + (q-1)\pi_2^{(k^\star)} + \cdots + (q + 1 - n_{k^\star})\pi_{n_{k^\star}}^{(k^\star)} = \sum_{j=1}^{n_{k^\star}} (q - j + 1)\pi_j - \Theta \sum_{j=1}^{n_{k^\star}} (q - j + 1).$$

Hence, the total capacity, $\sum_{e \in E(K_n)} u_e$, equals:

$$\begin{aligned}
&\sum_{j=1}^{n_{k^\star}} (q - j + 1)\pi_j - \Theta \sum_{j=1}^{n_{k^\star}} (q - j + 1) + \frac{q(q+1)}{2}\Theta \\
&\quad = \sum_{j=1}^{n_{k^\star}} (q - j + 1)\pi_j + \Theta \sum_{j=n_{k^\star}+1}^{q} (q - j + 1) \\
&\quad = \sum_{j=1}^{n_{k^\star}} (q - j + 1)\pi_j + \Theta \sum_{j=1}^{q-n_{k^\star}} j \\
&\quad = \sum_{j=1}^{n_{k^\star}} (q - j + 1)\pi_j + \frac{q - n_{k^\star}}{2}\, \xi_{n_{k^\star}+1} \\
&\quad = \sum_{j=1}^{q-p+1} (q - j + 1)\pi_j + \frac{(p-1)}{2}\, \xi_{q-p+2} \quad \text{for } p = (q + 1 - n_{k^\star}) \\
&\quad = LB_p^{(q)}(\pi).
\end{aligned}$$

Thus, we have shown that the algorithm produces a solution whose total capacity equals one of the stated bounds and hence is the minimum value possible. □

Note that the edge capacities in the synthesized network are not guaranteed to be integral. The paper on this integer version of the synthesis problem is under preparation [11].

References

[1] C.C. Aggarwal, J.B. Orlin, On multiroute maximum flows in networks, Networks 39 (2002) 43–52.

[2] W. Chou, H. Frank, Survivable communication networks and the terminal capacity matrix, IEEE Trans. Circuit Theory CT-17 (1970) 192–197.

[3] L.R. Ford, D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.
[4] A. Frank, Connectivity augmentation problems in network design, in: J. Birge, K.G. Murty (Eds.), Mathematical Programming: State of the Art, The University of Michigan Press, Michigan, 1994, pp. 34–63.
[5] R.J. Gibbens, F.P. Kelly, Dynamic routing in fully connected networks, IMA J. Math. Control Inform. 7 (1990) 77–111.
[6] R.E. Gomory, T.C. Hu, Multi-terminal network flows, J. Soc. Indust. Appl. Math. 9 (1961) 551–570.
[7] R.E. Gomory, T.C. Hu, An application of generalized linear programming to network flows, J. Soc. Indust. Appl. Math. 10 (1961) 260–283.
[8] R. Hassin, A. Levin, Synthesis of 2-commodity flow networks, Proceedings of IPCO, Lecture Notes in Computer Science, Vol. 2081, Springer, Berlin, 2001, pp. 226–235.
[9] M. Hojati, The network synthesis problem in a cycle, Oper. Res. Lett. 17 (1995) 231–236.
[10] S.N. Kabadi, R. Chandrasekaran, K.P.K. Nair, 2-commodity integer network synthesis problem, Working Paper, Faculty of Administration, University of New Brunswick, April, 2003.
[11] S.N. Kabadi, R. Chandrasekaran, K.P.K. Nair, Y.P. Aneja, Integer version of the multipath flow network synthesis problem, Working Paper, Faculty of Administration, University of New Brunswick, January, 2003.
[12] S.N. Kabadi, J. Kang, R. Chandrasekaran, K.P.K. Nair, Hop constrained network flows: analysis and synthesis, Working Paper, Faculty of Administration, University of New Brunswick, August, 2003.
[13] W. Kishimoto, A method for obtaining the maximum multi-route flows in a network, Networks 27 (1996) 279–291.
[14] W. Kishimoto, M. Takeuchi, On two-route flows in an undirected network, IEICE Technical Report, CAS90-19, DSP-90-23, 1990 (in Japanese).
[15] W. Kishimoto, M. Takeuchi, On *m*-route flows in a network, IEICE Trans. J. 76-A (1993) 1185–1200 (in Japanese).
[16] W. Kishimoto, M. Takeuchi, G. Kishi, Two-route flows in an undirected network, IEICE Trans. J. 75-A (1992) 1699–1717 (in Japanese).
[17] W. Mayeda, Terminal and branch capacity matrices of a communication net, IRE Trans. Circuit Theory CT-7 (1960) 261–269.
[18] R. Sridhar, R. Chandrasekaran, Integer solution of synthesis of communication network, Math. Oper. Res. 17 (3) (1992) 581–585.
[19] K. Talluri, Network synthesis with few edges, Networks 27 (1996) 109–115.